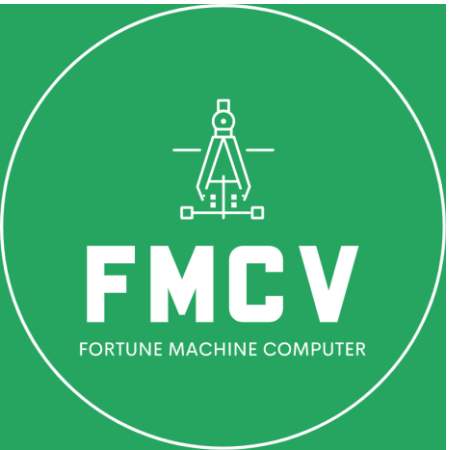
Segment Anything 2

20240812 SAM 2: Segment Anything in Images and Videos Inference and Fine Tuning Training

Email: chong@fmcv.my



SAM 2

Install python 3.12 and run in command prompt terminal

pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu124

git clone https://github.com/facebookresearch/segment-anything-2

cd segment-anything-2; pip install -e .

Download model

- sam2_hiera_small.pt
  https://dl.fbaipublicfiles.com/segment_anything_2/072824/sam2_hiera_small.pt

- sam2_hiera_large.pt
  https://dl.fbaipublicfiles.com/segment_anything_2/072824/sam2_hiera_large.pt

- Put into segment-anything-2/checkpoints folder

Run inferences

in command prompt terminal

pip install jupyterlab

Jupyter-lab

In internet browser

segment-anything-2/notebooks/video_predictor_example.ipynb

Step 3: Propagate the prompts to get the masklet across the video

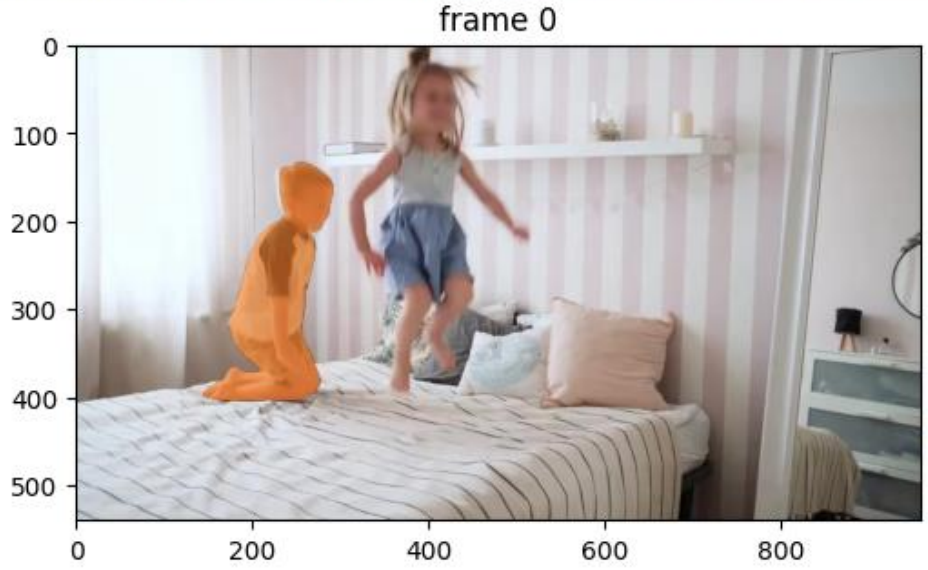
To get the masklet throughout the entire video, we propagate the prompts using the `propagate_in_video` API.

In [13]:

```
# run propagation throughout the video and collect the results in a dict
video_segments = {}  # video_segments contains the per-frame segmentation results
for out_frame_idx, out_obj_ids, out_mask_logits in predictor.propagate_in_video(inference_state):
    video_segments[out_frame_idx] = {
        out_obj_id: (out_mask_logits[i] > 0.0).cpu().numpy()
        for i, out_obj_id in enumerate(out_obj_ids)
    }

# render the segmentation results every few frames
vis_frame_stride = 15
plt.close("all")
for out_frame_idx in range(0, len(frame_names), vis_frame_stride):
    plt.figure(figsize=(6, 4))
    plt.title(f"frame {out_frame_idx}")
    plt.imshow(Image.open(os.path.join(video_dir, frame_names[out_frame_idx])))
    for out_obj_id, out_mask in video_segments[out_frame_idx].items():
        show_mask(out_mask, plt.gca(), obj_id=out_obj_id)
```

```
propagate in video: 100%|██████████| 200/200 [00:08<00:00, 23.76it/s]
```



Fine tunning/Train

- Tutorial : https://medium.com/@sagieppel/train-fine-tune-segment-anything-2-sam-2-in-60-lines-of-code-928dd29a63b3
- Git clone https://github.com/cyysky/fine-tune-train_segment_anything_2_in_60_lines_of_code.git
- Download and Extract https://zenodo.org/records/3697452/files/LabPicsV1.zip?download=1
- Jupyter-lab
- Run in browser
- fine-tune-train_segment_anything_2_in_60_lines_of_code/20240811_Inference_SAM_2.ipynb

In [8]:
```
scaler = torch.cuda.amp.GradScaler() # set mixed precision
```

In [9]:
```
for itr in range(100000):
    with torch.cuda.amp.autocast(): # cast to mix precision
        #with torch.cuda.amp.autocast():
            image,mask,input_point, input_label = read_batch(data) # load data batch
            if mask.shape[0]==0: continue # ignore empty batches
            predictor.set_image(image) # apply SAM image encodet to the image

            # prompt encoding

            mask_input, unnorm_coords, labels, unnorm_box = predictor._prep_prompts(input_point, input_label, box=None, mas
            sparse_embeddings, dense_embeddings = predictor.model.sam_prompt_encoder(points=(unnorm_coords, labels),boxes=N

            # mask decoder

            batched_mode = unnorm_coords.shape[0] > 1 # multi object prediction
            high_res_features = [feat_level[-1].unsqueeze(0) for feat_level in predictor._features["high_res_feats"]]
            low_res_masks, prd_scores, _, _ = predictor.model.sam_mask_decoder(image_embeddings=predictor._features["image_
            prd_masks = predictor._transforms.postprocess_masks(low_res_masks, predictor._orig_hw[-1])# Upscale the masks t

            # Segmentaion Loss caclulation

            gt_mask = torch.tensor(mask.astype(np.float32)).cuda()
            prd_mask = torch.sigmoid(prd_masks[:, 0])
            seg_loss = (-gt_mask * torch.log(prd_mask + 0.00001) - (1 - gt_mask) * torch.log((1 - prd_mask) + 0.00001)).mea

            # Score loss calculation (intersection over union) IOU

            inter = (gt_mask * (prd_mask > 0.5)).sum(1).sum(1)
            iou = inter / (gt_mask.sum(1).sum(1) + (prd_mask > 0.5).sum(1).sum(1) - inter)
            score_loss = torch.abs(prd_scores[:, 0] - iou).mean()
            loss=seg_loss+score_loss*0.05  # mix losses

            # apply back propogation

            predictor.model.zero_grad() # empty gradient
            scaler.scale(loss).backward()  # Backpropogate
            scaler.step(optimizer)
            scaler.update() # Mix precision

            if itr%1000==0: torch.save(predictor.model.state_dict(), "model.torch") # save model

            # Display results

            if itr==0: mean_iou=0
            mean_iou = mean_iou * 0.99 + 0.01 * np.mean(iou.cpu().detach().numpy())
            print("step)",itr, "Accuracy(IOU)=",mean_iou)
```

Inference fine-tunned models

- Jupyter-lab
- fine-tune-train_segment_anything_2_in_60_lines_of_code/blob/main/20240811_Inference_SAM_2.ipynb

In [7]:
```
# Build net and load weights
predictor = SAM2ImagePredictor(sam2_model)
predictor.model.load_state_dict(torch.load("l_model_6047.torch"))
```

Out[7]:
```
<All keys matched successfully>
```

In [8]:
```
# predict mask

with torch.no_grad():
    predictor.set_image(image)
    masks, scores, logits = predictor.predict(
        point_coords=input_points,
        point_labels=np.ones([input_points.shape[0],1])
    )
```

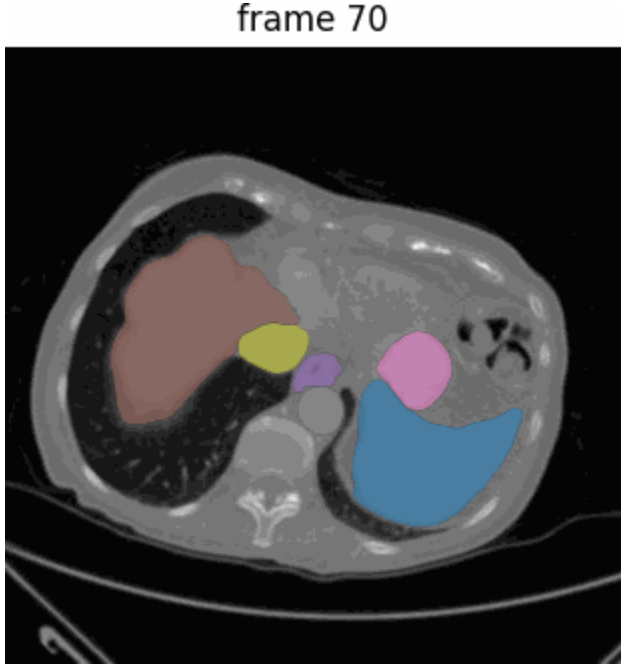
```
D:\workspace\20240810_Segment_Anything_2\segment-anything-2\sam2\modeling\backbones\hieradet.py:68: UserWarning: 1Torch was n
ot compiled with flash attention. (Triggered internally at ..\aten\src\ATen\native\transformers\cuda\sdp_utils.cpp:455.)
  x = F.scaled_dot_product_attention(
```

In [12]:
```
# Short predicted masks from high to low score

np_masks = np.array(masks[:,0])
np_scores = scores[:,0]
shorted_masks = np_masks[np.argsort(np_scores)][::-1]
```

Use cases

- Medical SAM 2: Segment Medical Images As Video Via Segment Anything Model 2 : https://arxiv.org/abs/2408.00874
- Git clone https://github.com/MedicineToken/Medical-SAM2



Resources

- Introducing SAM 2: The next generation of Meta Segment Anything Model for videos and images : https://ai.meta.com/blog/segment-anything-2/
- Paper : https://ai.meta.com/research/publications/sam-2-segment-anything-in-images-and-videos/
- Meta SA-V Datasets : https://ai.meta.com/datasets/segment-anything-video/
- Train/Fine-Tune : https://medium.com/@sagieppel/train-fine-tune-segment-anything-2-sam-2-in-60-lines-of-code-928dd29a63b3